

The Policies of Designing Differentially Private Mechanisms: Utility First vs. Privacy First

Genqiang Wu^{1,2}, Xianyao Xia¹, and Yeping He^{1,3}

¹ NFS, Institute of Software Chinese Academy of Sciences, Beijing 100190, China
genqiang80@gmail.com, {xianyao,yeping}@nfs.iscas.ac.cn

² SIE, Lanzhou University of Finance and Economics, Lanzhou 730020, China

³ SKLCS, Institute of Software Chinese Academy of Sciences, Beijing 100190, China

Abstract. Sensitivity-based methods are extensively used to construct differentially private mechanisms. In this paper, we realize that designing a differentially private mechanism can be considered as finding a randomized mapping between two metric spaces. The metric in the first metric space can be considered as a metric about privacy and the metric in the second metric space can be considered as a metric about utility. We find that the sensitivity-based methods are those just using the metric about utility to construct mechanisms. By the observation, we design mechanisms based on the metric about privacy. Furthermore, we design mechanisms based on the composition of the two metrics. Moreover, we find that most mechanisms, such as the global sensitivity mechanism [1], the Staircase mechanism [2,3], the Ladder mechanism [4] and the K -norm mechanism [5,6], can be considered as special cases of our mechanisms. Finally, we analyze these mechanisms and apply them to the subgraph counting problem and the linear query problem. The experiments show that our mechanisms (in most cases) have more accurate results than the state of the art mechanisms.

Keywords: differential privacy, mechanism design, abstract model, metric space, sensitivity method

1 Introduction

Differential privacy studies how to query dataset while preserving the privacy of individuals whose sensitive informations are contained in the dataset. The key of differential privacy is to find efficient algorithms (or in other words, mechanisms) to query sensitive dataset to obtain (relatively) accurate outputs while satisfying differential privacy. The global sensitivity-based method [1] is an efficient and widely used method to achieve differential privacy. Explicitly, the global sensitivity of a query function is its maximum difference when evaluated on any two neighbouring datasets (differing on at most one record), and the global sensitivity-based method is to perturb the query result with noise of magnitude proportional to the global sensitivity [1]. Although the global sensitivity-based method is powerful, it works poorly when the global sensitivity is large.

Due to this reason, there are a lot of works to improve the global sensitivity-based method, such as the smooth sensitivity-based method [7] and the local sensitivity-based method [4] among other variants or approximation methods [8,9,10,11], where the local sensitivity of a query f on a dataset x is the maximum difference of $f(x)$ with the value of its any neighboring dataset, and the smooth sensitivity is a compromise between the global sensitivity and the local sensitivity. The (global or local) sensitivity has long been seen as a mark of the noise magnitude needed in order to satisfy differential privacy in practice. However, there is seldom theoretical result to support this point. Our questions are: Does there exist query function which has very large (global or local) sensitivity but has (relatively) accurate differentially private query results? Does there exist any other feature(s) which can characterize the noise magnitude needed to satisfy differential privacy. Are there other universal methods to design differentially private mechanisms except the sensitivity-based methods.

To answer these question, we first need to understand differential privacy in a simple way. However, since differential privacy model is task-specific, i.e., each query function needs a specific differentially private mechanism, which makes it hard to find common features to characterize all or most of differential privacy problems [12,5,13,14,15,16,9]. Due to this reason, we should first formalize differential privacy problems in a universal way.

1.1 Contribution

In this paper, we focus on how to design universal differentially private mechanisms. The contributions are as follows.

We first give an abstract model to formalize differential privacy problems in a universal way. In this model a differential privacy problem is modeled as finding a randomized mapping between two metric spaces. The privacy is modeled as the property of the randomized mapping to the first metric space and the utility is modeled as the property of the randomized mapping to the second metric space.

The above formalization makes it clear to understand most differential privacy problems. We reinterpret most of the sensitivity-based mechanisms, including the global sensitivity-based mechanisms [1,2,3], the local sensitivity-based mechanism [4] etc., in a uniform way. These mechanisms can be considered as the mechanisms designed by using the second metric, which can be explained as the utility-oriented mechanisms. We then design the mechanisms which are based on the first metric and can be explained as the privacy-oriented mechanisms. We find that the K -norm mechanism [5,6] can be considered as a special case of the later kind of mechanisms. Furthermore, we also construct several mechanisms by using the combination of the two metrics.

Finally, we analyze our mechanisms and apply them to the subgraph counting problem and the linear query problem. The experiments show that our mechanisms (in most cases) have more accurate results than the state of the art mechanisms.

1.2 Outline

The rest of the paper is organized as follows: Section 2 presents some fundamental materials. Section 3 gives an abstract model of differential privacy, which formalizes a differential privacy problem as finding a randomized mapping between two metric spaces. In Section 4, we construct six mechanisms by using the two metrics of the two metric spaces. In Section 5, we analyze the utility of our mechanisms. In Section 6 we apply the mechanisms in Section 4 to the subgraph counting problem and the linear query problem and give some experiments. Section 7 presents the related works. Finally, concluding remarks and a discussion of future works are presented in Section 8.

2 Preliminaries

2.1 Notations

Let $\bar{\mathbb{N}}$ denote the set of the natural numbers including 0. Let $\|z\|_p$ represent the ℓ_p -norm of z . Let $\bar{\mathbb{Q}} = \mathbb{R} - \mathbb{Q}$. In this paper, unless noted otherwise, any set is *not* a multiset.

2.2 Differential Privacy

A dataset is a collection (a multiset) of n records, each of which is derived from the record universe \mathcal{X} and denotes the information of one individual. We use the histogram representation $x \in \bar{\mathbb{N}}^{|\mathcal{X}|}$ to denote the dataset x , where the i th entry x_i of x represents the number of elements in x of type $i \in \mathcal{X}$ [12,17,18]. Two datasets x, x' are said to be *neighbors* (or *neighboring datasets*) if $\|x - x'\|_1 = 1$.

Differential privacy [19,17] characterizes privacy problems by capturing the changes of outputs when one's record in a dataset is changed. The changes of datasets are captured by the notion of the neighboring datasets. For the dataset universe \mathcal{D} and a query function f , let $\mathcal{R} = \{f(x) : x \in \mathcal{D}\}$ and equip \mathcal{R} with a Borel σ -algebra \mathcal{B} .

Definition 1 ((ϵ, δ)-Differential Privacy). *For the dataset universe \mathcal{D} , let $\mathcal{P}(\mathcal{R})$ denote the set of all the probability measures on $(\mathcal{R}, \mathcal{B})$. A mapping $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{P}(\mathcal{R})$ gives (ϵ, δ)-differential privacy if for all neighbors $x, x' \in \mathcal{D}$, and all $S \in \mathcal{B}$,*

$$\Pr[\mathcal{M}(x) \in S] \leq e^\epsilon \Pr[\mathcal{M}(x') \in S] + \delta, \quad (1)$$

where we abuse the notation $\mathcal{M}(x)$ as either denoting a probability distribution in $\mathcal{P}(\mathcal{R})$ or denoting a random variable following the probability distribution. If \mathcal{M} satisfies ($\epsilon, 0$)-differential privacy, we say that \mathcal{M} satisfies ϵ -differential privacy.

2.3 Achieving Differential Privacy

The global sensitivity-based method is a basic method to achieve differential privacy [1]. We first define the global sensitivity.

Definition 2 (Global Sensitivity). *For the query function $f : \mathcal{D} \rightarrow \mathcal{R}$, if $\mathcal{R} \subseteq \mathbb{R}^k$, then the global sensitivity of f is defined as*

$$\Delta f = \max_{x, x' \in \mathcal{D}: \|x - x'\|_1 = 1} \|f(x) - f(x')\|_1.$$

Otherwise, letting $q^x : \mathcal{R} \rightarrow (-\infty, 0]$ be the score function when inputting the dataset x and outputting the point $r \in \mathcal{R}$, the global sensitivity of q^x is defined as

$$\Delta q = \max_{r \in \mathcal{R}, x, x' \in \mathcal{D}: \|x - x'\|_1 = 1} |q^x(r) - q^{x'}(r)|.$$

The Laplace mechanism [19,17] is one important global sensitivity-based mechanism and the Exponential mechanism [20] is one generalization of the global sensitivity-based mechanisms.

Definition 3. *The Laplace mechanism $\mathcal{M}(x, f)$ generates a real random vector $r = (r_1, \dots, r_k)$ with density*

$$p^x(r) = \frac{\epsilon}{2\Delta f} \exp\left(-\frac{\epsilon \|r - f(x)\|_1}{\Delta f}\right).$$

The Exponential mechanism $\mathcal{M}(x, q)$ outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp(\frac{\epsilon q^x(r)}{2\Delta q})$.

Both the Laplace mechanism and the Exponential mechanism satisfy ϵ -differential privacy [17].

3 The Abstract Model of Differential Privacy

In this section, we give an abstract model of differential privacy, the intention of which is to reinterpret and formalize differentially private data processing problems in a universal way. There are somewhat similar treatments in [21,22].

3.1 The Dataset Metric Space and the Value Metric Space

The dataset universe is modeled as a set \mathcal{D} on which a metric \bar{d} is defined.

Definition 4 (Dataset Metric Space). *Let f be a function defined on the set \mathcal{D} on which a metric \bar{d} is defined. Then the metric space (\mathcal{D}, \bar{d}) is called the dataset metric space of f . Two elements $x, y \in \mathcal{D}$ are said to be neighbors (or neighboring datasets) of distance k if $k - 1 < \bar{d}(x, y) \leq k$, for $k \in \mathbb{N}$. When $k = 1$, x, y are said to be neighbors.*

We set $\mathcal{N}_i^x = \{y \in \mathcal{D} : i-1 < \bar{d}(x, y) \leq i\}$, for $i \in \bar{\mathbb{N}}$. We set $\bar{\mathcal{N}}_i^x = \{y \in \mathcal{D} : \bar{d}(x, y) \leq i\}$ for $i \in \bar{\mathbb{N}}$ and set $\mathcal{N}^x = \{y \in \mathcal{D} : \bar{d}(x, y) \leq 1\}$ for abbreviation. The codomain of a query function f on \mathcal{D} is modeled as a set \mathcal{R} over which a metric d is defined.

Definition 5 (Value Metric Space). For a function f on \mathcal{D} , set $\mathcal{R} = \{f(x) : x \in \mathcal{D}\}$. Defining a metric d on \mathcal{R} , then (\mathcal{R}, d) is called the value metric space of f . Equipping \mathcal{R} with the Borel σ -algebra \mathcal{B} generated by the open sets in \mathcal{R} (in the metric topology), then $(\mathcal{R}, \mathcal{B})$ is a measurable space.

The product metric space and the product probability space are used to model the batch query functions.

Definition 6 (Product Metric Space). If $(\mathcal{R}_1, d_1), \dots, (\mathcal{R}_n, d_n)$ are metric spaces, and N is a norm on \mathbb{R}^n , then $(\mathcal{R}_1 \times \dots \times \mathcal{R}_n, N(d_1, \dots, d_n))$ is a metric space, where the product metric $N(d_1, \dots, d_n)$ is defined by

$$N(d_1, \dots, d_n)((x_1, \dots, x_n), (y_1, \dots, y_n)) = N(d_1(x_1, y_1), \dots, d_n(x_n, y_n)),$$

and the induced topology agrees with the product topology.

Definition 7 (Product Probability Space). Let $(\mathcal{R}_1, \mathcal{B}_1, \mu_1), \dots, (\mathcal{R}_n, \mathcal{B}_n, \mu_n)$ be n probability spaces. Then the probability space $(\mathcal{R}, \mathcal{B}, \mu)$, defined by $\mathcal{R} = \mathcal{R}_1 \times \dots \times \mathcal{R}_n$, $\mathcal{B} = \mathcal{B}_1 \times \dots \times \mathcal{B}_n$ and $\mu = \mu_1 \times \dots \times \mu_n$, is called the product probability space of the n probability spaces.

For n query functions f_1, \dots, f_n over the dataset metric space (\mathcal{D}, \bar{d}) , let $(\mathcal{R}_1, d_1), \dots, (\mathcal{R}_n, d_n)$ be their value metric spaces respectively. Then the product metric space $(\mathcal{R}_1 \times \dots \times \mathcal{R}_n, N(d_1, \dots, d_n))$ is called the (product) value metric space of f_1, \dots, f_n .

3.2 The Definition of Differential Privacy

Definition 8 (ϵ -Differential Privacy). For a dataset metric space (\mathcal{D}, \bar{d}) , let $\mathcal{P}(\mathcal{R})$ denote the set of all the probability measures on $(\mathcal{R}, \mathcal{B})$. A mapping $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{P}(\mathcal{R})$ gives ϵ -differential privacy if for all neighbors $x, x' \in \mathcal{D}$, and all $S \in \mathcal{B}$,

$$\Pr[\mathcal{M}(x) \in S] \leq e^\epsilon \Pr[\mathcal{M}(x') \in S], \quad (2)$$

where we abuse the notation $\mathcal{M}(x)$ as either denoting a probability distribution in $\mathcal{P}(\mathcal{R})$ or denoting a random variable following the probability distribution.

Proposition 1 (Composition Privacy). For the dataset metric space (\mathcal{D}, \bar{d}) , let \mathcal{M}_i be ϵ_i -differentially private on $(\mathcal{R}_i, \mathcal{B}_i)$ for $i \in \{1, \dots, n\}$. Then the composition of $\mathcal{M}_1, \dots, \mathcal{M}_n$, defined by $\mathcal{M}(x) = (\mathcal{M}_1(x), \dots, \mathcal{M}_n(x))$, $x \in \mathcal{D}$, is $\sum_{i=1}^n \epsilon_i$ -differentially private on the product probability space $(\mathcal{R}, \mathcal{B})$.

Proof. The proof is similar with the one of Theorem 3.14 in [17] and is omitted.

Proposition 2 (Group Privacy). *Let \mathcal{M} be an ϵ -differentially private mechanism. Assume that, for any $x, y \in \mathcal{D}$ with $\bar{d}(x, y) = i$ for $i \in \mathbb{N}$, there exists $x' \in \mathcal{D}$ such that $\bar{d}(x, x') = 1$ and $\bar{d}(x', y) = i - 1$. Then*

$$\Pr[\mathcal{M}(x) \in S] \leq e^{i\epsilon} \Pr[\mathcal{M}(y) \in S],$$

for any $S \in \mathcal{B}$.

Proof. The proposition is an immediate corollary of the inequality (2).

3.3 Utility Model

Let (\mathcal{R}, d) be the value metric space of f . Set $C_T^x = \{r \in \mathcal{R} : d(f(x), r) \leq T\}$ for all $x \in \mathcal{D}$, where $T > 0$. We use the probability P_T^x to measure the utility (or accuracy) of \mathcal{M} at the dataset x , where

$$P_T^x = \Pr[\mathcal{M}(x) \in C_T^x] = \int_{r \in C_T^x} p^x(r) \mu(dr),$$

and where $p^x(r)$ is the probability distribution function of $\mathcal{M}(x)$, μ is a measure on $(\mathcal{R}, \mathcal{B})$. We use the set $\{P_T^x : x \in \mathcal{D}\}$ to measure the accuracy of \mathcal{M} . An alternative to measure the utility of \mathcal{M} at x is to use the expected value of the distance $d(\mathcal{M}(x), f(x))$, i.e.,

$$\mathbb{E}[d(\mathcal{M}(x), f(x))] = \int_{r \in \mathcal{R}} d(r, f(x)) p^x(r) \mu(dr).$$

Note that the accuracy of the batch queries is measured on their product value metric space.

3.4 Query Function

Linear function is known to be one kind of the simplest query functions in differential privacy, which is a generalization of the sum function or the counting function.

Definition 9 (Linear Function). *For the query function $f : \mathcal{D} \rightarrow \mathcal{R}$, assume both of \mathcal{D}, \mathcal{R} are vector spaces [23]. If $f(x + y) = f(x) + f(y)$ for all $x, y \in \mathcal{D}$, the function f is said to be a linear (query) function.*

Since for a linear function f , the set

$$\{f(x') - f(x) : x' \in \mathcal{N}^x\} = \{f(x' - x) : x' \in \mathcal{N}^x\} = \{f(y) : y \in \mathcal{D} \wedge \bar{d}(0, y) \leq 1\}$$

has no difference for different x and seems to be different from other linear queries, we will represent f by the set. We denote $\mathcal{V}_f = \{f(x') - f(x) : x' \in \mathcal{N}^x\}$ and call it *the neighboring set* of f . Any query function, which is not a linear function, is said to be a non-linear (query) function.

Definition 10 (Monotonic Function). *The function f is said to be a monotonic (query) function if for any $x \in \mathcal{D}$ and all $y, z \in \mathcal{D}$ satisfying $\bar{d}(x, y) > \bar{d}(x, z)$, there is $d(f(x), f(y)) \geq d(f(x), f(z))$.*

Definition 11 (Permutation Function and Identity Function). *An injective function f is called a permutation function if its value metric space (\mathcal{R}, d) is the same as its dataset metric space (\mathcal{D}, \bar{d}) . Moreover, if $f(x) = x$ for all $x \in \mathcal{D}$, then f is called an identity function.*

The identity function is used to model the data publication problem in differential privacy [24,25].

Definition 12 (Global Sensitivity and Local Sensitivity). *Let (\mathcal{D}, \bar{d}) , (\mathcal{R}, d) be the dataset metric space, the value metric space of the function f , respectively. The global sensitivity of f is defined as*

$$\Delta f = \max_{x, x' \in \mathcal{D}: \bar{d}(x', x) \leq 1} d(f(x), f(x')).$$

The local sensitivity of f at x is defined as

$$\Delta_f^x = \max_{x' \in \mathcal{D}: \bar{d}(x', x) \leq 1} d(f(x), f(x')).$$

The local sensitivity of distance i of f at x is defined as

$$\Delta_i^x = \Delta_{i,f}^x = \max_{x' \in \mathcal{D}: \bar{d}(x', x) \leq i} \Delta_f^{x'}.$$

Obviously, $\Delta_f^x = \Delta_0^x$.

Note that the definitions of the global sensitivity and the local sensitivity are consistent with those in [1,7,4].

3.5 Instance Interpretation

The abstract model in Section 3 is consistent with the classic differential privacy model. In this section, we give some instance interpretations. First, for a query function, the set \mathcal{D} in the dataset metric space (\mathcal{D}, \bar{d}) is equivalent to the dataset universe in the classic differential privacy model. The metric \bar{d} captures the mathematical meaning of the neighboring relation of datasets. The details are as follows. There are two different definitions about neighboring datasets in differential privacy: *bounded neighboring datasets* and *unbounded neighboring datasets* [26]. For the definition of bounded neighboring datasets, all of the datasets are assumed to have the same number n of records. Two datasets $x, x' \in \mathcal{D}$ are said to be neighboring datasets if $\|x - x'\|_1 = 2$, where x, x' are their histogram representations as noted in Section 2.2. In this case, we can set $\bar{d}(x, y) = \frac{\|x - y\|_1}{2}$. For the definition of unbounded neighboring datasets, the number of records in a dataset can be any natural number. Two datasets

$x, x' \in \mathcal{D}$ are said to be neighboring datasets if $\|x - x'\|_1 = 1$. In this case, we can set $\bar{d}(x, y) = \|x - y\|_1$.

Theoretically, in differential privacy, almost all of data processing problems can be explained as a function f whose domain is set to be \mathcal{D} and whose codomain is set to be \mathcal{R} , such as the SQL query problems [27,9,28,5], the statistical problems [29,30], and the data mining and machine learning problems [31,32,33,34,35,36,14,15,16]. The idea of differential privacy to preserve privacy can be explained as follows: When the real dataset is x , in order to preserve privacy, a differentially private mechanism first samples a dataset $y \in \mathcal{D}$ (according to a probability distribution) and then outputs $f(y)$ as the final query result of f . There should be a distortion function $L^x(r)$ to measure the distortion when querying the dataset x but outputting $f(y)$. Then, the metric d of the value metric space (\mathcal{R}, d) can be set as

$$d(f(x), r) = L^x(r), \quad x \in \mathcal{D}, r \in \mathcal{R}, \quad (3)$$

which is a measure of the distortion of querying the dataset x but outputting r . For example, if $\mathcal{R} \subseteq \mathbb{R}^k$, we can set $d(f(x), r) = \|f(x) - r\|_p$ [12,18]; if \mathcal{R} is a set of real matrices, we can use a norm over matrices to define d [13,14,37]. For the case of \mathcal{R} being a set of non-numeric elements, the corresponding distortion function $L^x(r)$, in general, will not satisfy the triangular inequality property and the symmetric property of metric [32,24,25]. In this condition, the metric d can be considered as an ideal approximation of $L^x(r)$ and we would treat the real problem by the method found when treating the ideal problem, which would simplify the complexity of complex problems. We now give some examples.

Example 1 (counting query). A counting query function f outputs non-negative integers. Then, we can set $\mathcal{R} = \{0, \dots, k\}$, and $d(r, f(x)) = |r - f(x)|$. The subgraph counting query is a special kind of counting query [38,9,10,27].

Example 2 (multi-linear queries [12,17,18,39]). For k real valued linear queries f_1, \dots, f_k , we can set $f(x) = (f_1(x), \dots, f_k(x))$ for all $x \in \mathcal{D}$. Then \mathcal{R} can be set as a subset of \mathbb{R}^k and $d(r, f(x)) = \|r - f(x)\|_p$.

Example 3 (data publishing [32,24,25]). For the data publishing problem, the query function can be defined as the identity function as defined in Definition 11 where the codomain \mathcal{R} of f is the same as its domain \mathcal{D} . There will be different way to set the metric d , the simplest way is to set $d = \bar{d}$, i.e., the metric induced by the ℓ_1 -norm.

Example 4 (principal component analysis [13,14,37]). For the principal component analysis problem, each record is a real-valued vector and a dataset x is an $n \times m$ real-valued matrix. The value $f(x)$ is the principal component analysis matrix of x , i.e., a $k \times m$ real-valued matrix. Then \mathcal{R} is a set of $k \times m$ real-valued matrices. The metric $d(f(x), r)$ can be set as the spectral norm or the frobenius norm of the matrix $r - f(x)$.

Example 5 (linear classifier [36,40,15]). For the linear classifier problem, the value $f(x)$ can be set as the classifier of the dataset x (if there are several candidates, just choose one randomly), i.e., a k dimensional real-valued vector, which is the output of a classifier algorithm, such as the logistic regression algorithm. Then \mathcal{R} is a set of k dimensional real-valued vectors and $d(r, f(x)) = \|r - f(x)\|_p$.

Example 6 (functional output [41,36,42]). As noted in [41], there are many applications, where the outputs are functions, such as the density functions. In this case, the value of $f(x)$ would be a function and the codomain \mathcal{R} would be a set of functions. Let $(\mathcal{F}, \|\cdot\|)$ be a normed space [23] and let $\mathcal{R} \subseteq \mathcal{F}$, then $d(r, f(x)) = \|r - f(x)\|$.

3.6 Discussion

By the formalization of Section 3, we can reinterpret differential privacy as follows. The basic idea of differential privacy to preserve privacy is: When the real dataset is x , in order to preserve privacy, a privacy mechanism first samples a dataset $y \in \mathcal{D}$ (according to a probability distribution) and then outputs $f(y) \in \mathcal{R}$ as the final query result of f . Or equivalently, the privacy mechanism directly samples a value r from the codomain \mathcal{R} of f as the final query result. Note that the above privacy mechanism should be query-specific since the codomain \mathcal{R} and the distortion function $L^x(r)$ of f are different from other query functions', in general.

A differentially private mechanism \mathcal{M} of a function f is to find a randomized mapping \mathcal{M} from the dataset metric space (\mathcal{D}, \bar{d}) to its value metric space (\mathcal{R}, d) . When inputting x , it outputs a probability distribution $\mathcal{M}(x) \in \mathcal{P}(\mathcal{R})$, or equivalently, outputs a random variable $\mathcal{M}(x)$ following the probability distribution as defined in Definition 8. Differential privacy needs the mapping \mathcal{M} should *control the upper bound* of the distance of outputs according to the distance of inputs. Specifically, for two different inputs $x, y \in \mathcal{D}$, the distance of the outputs $\mathcal{M}(x), \mathcal{M}(y)$ is upper bounded by the inequality

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\bar{d}(x,y)\epsilon} \Pr[\mathcal{M}(y) \in S], \quad (4)$$

which can be considered as a distance measure of two probability distributions $\mathcal{M}(x), \mathcal{M}(y)$. (Of course, there should be other distance measures to $\mathcal{M}(x), \mathcal{M}(y)$, such as those in [43,44].) On the other hand, the utility needs the mapping \mathcal{M} should, when inputting $x \in \mathcal{D}$, output a random variable $\mathcal{M}(x)$ which is a good approximation to $f(x)$. That is, the random variable $\mathcal{M}(x)$ should be a 'differentially private' approximation of $f(x)$. This implies that high utility means that the outputs of the mapping \mathcal{M} should *keep* the distance of the metric space (\mathcal{R}, d) . That is, if $d(f(x), f(y))$ is large, then the two random variables $\mathcal{M}(x), \mathcal{M}(y)$ should also have large distance in order to keep high utility. Note that there would be a contradiction if both high privacy and high utility need holding when $\bar{d}(x, y)$ is small but $d(f(x), f(y))$ is large. Therefore, there should be a tradeoff between privacy and utility. Furthermore, there is another kind of

tradeoff in differential privacy which balances the utility of \mathcal{M} among different datasets in \mathcal{D} . In the next section we will give some concrete solutions for these tradeoffs.

Note that the differential privacy problem presented above is different from the randomized metric embedding problem in [45] where one needs to find a randomized mapping \mathcal{M} between two metric spaces (\mathcal{D}, \bar{d}) , (\mathcal{R}, d) such that, for any $x, y \in \mathcal{D}$, there is

$$\bar{d}(x, y)/c \leq d(\mathcal{M}(x), \mathcal{M}(y)) \leq \bar{d}(x, y) \quad (5)$$

with probability $1 - \eta$. Although both of them treat the randomized mapping problem between two metric spaces, there are several major differences. First, in the differential privacy there is one query function f between two metric spaces but none in the randomized metric embedding. Second, in the randomized metric embedding the randomized mapping \mathcal{M} should keep the distance of the first metric space by the inequality (5). However, in the differential privacy, the privacy requirement needs the randomized mapping \mathcal{M} only *control the upper bound* of the distance of the outputs according to the distance of the inputs using the inequality (2), which implies that the case where the outputs of the randomized mapping are the same is allowable. Third, in the differential privacy the high utility *implies* the outputs of \mathcal{M} should keep the distance of the second spaces. However, the above implication is not reversible since there may be the case where the outputs of \mathcal{M} keeps the distance of the second metric but the utility is low.

One shortcoming of the abstract model of differential privacy is that there are many differential privacy problems whose utility functions are not metrics [24,25]. However, we believe that the model captures the core issue of differential privacy, i.e., the privacy-utility tradeoff. Figuring out the problems in the model would be the cornerstone to understand the whole field of differential privacy.

Note that, by the results of this section, any function f can have its own differential privacy problem so long as its domain and codomain both are metric spaces. Therefore, an SQL query function may have the same (or similar) differential privacy problem as a mathematical function. From this aspect we can say that it is the obligation not just of computer scientists but also of mathematicians to figure out differential privacy.

4 The Policies of Mechanism Design

In this section, we present our ideas to design differentially private mechanisms. We will first use the metrics \bar{d}, d interchangeably to design mechanisms. Then, we will use them combinatorially to design mechanisms.

4.1 The Basic Mechanisms

We first consider a simple mechanism presented in [2], which we called *Staircase mechanism*. Let $f : \mathcal{D} \rightarrow \mathbb{R}$ be a real-valued query function. The idea of the

Staircase mechanism is to set more high probability to those points in \mathbb{R} if they are more near to $f(x)$. This is reasonable since, for a point $r \in \mathbb{R}$, the distance $|r - f(x)|$ denotes the distortion extent when inputting x and outputting r . The Staircase mechanism is constructed as follows (with slight difference). First, for each dataset $x \in \mathcal{D}$, the codomain of f , i.e. the set \mathbb{R} , is partitioned into a set sequence $\{\mathcal{S}_i^x : i \in \bar{\mathbb{N}}\}$, where

$$\mathcal{S}_i^x = \{r \in \mathbb{R} : (i-1)\Delta f < |r - f(x)| \leq i\Delta f\}.$$

Second, set the probability distribution function $p^x(r)$ of $\mathcal{M}(x)$ as $p^x(r) = \frac{1}{\alpha} e^{-i\epsilon}$ to those $r \in \mathcal{S}_i^x$, where $\alpha = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{S}_i^x) = \Delta f \sum_{i=1}^{\infty} e^{-i\epsilon}$ is the normalization. One can verify that, for any two neighboring datasets x, x' , their corresponding set sequences $\{\mathcal{S}_i^x : i \in \bar{\mathbb{N}}\}$ and $\{\mathcal{S}_i^{x'} : i \in \bar{\mathbb{N}}\}$ satisfy that, for any two integers $s, t \in \bar{\mathbb{N}}$, $\mathcal{S}_s^x \cap \mathcal{S}_t^{x'} \neq \emptyset$ only when $|s - t| \leq 1$. Fig. 1 shows the geometric explanation. Therefore, the above mechanism satisfies ϵ -differential privacy. For general query functions, we formalize the Staircase mechanism as follow.

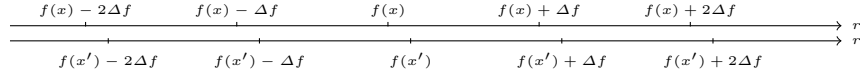


Fig. 1: Partition of \mathcal{R} in Staircase mechanism, where the upper, below lines show the set sequences $\{\mathcal{S}_i^x : i \in \bar{\mathbb{N}}\}$, $\{\mathcal{S}_i^{x'} : i \in \bar{\mathbb{N}}\}$, respectively

Theorem 1 (Staircase Mechanism). *For the dataset metric space (\mathcal{D}, \bar{d}) and a query function f , let (\mathcal{R}, d) be the value metric space of f . Let Δf be the global sensitivity of f . Set*

$$\mathcal{S}_i^x = \{r \in \mathcal{R} : (i-1)\Delta f < d(r, f(x)) \leq i\Delta f\}, \quad i \in \bar{\mathbb{N}}.$$

For the mechanism \mathcal{M} and a dataset $x \in \mathcal{D}$, let the probability distribution $p^x(r)$ of $\mathcal{M}(x)$ be

$$p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}, \quad \text{when } r \in \mathcal{S}_i^x,$$

where $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{S}_i^x)$. Then the mechanism \mathcal{M} satisfies 2ϵ -differential privacy.

Proof. Let $x, x' \in \mathcal{D}$ be two neighbours. One can verify that, for any two integers $s, t \in \bar{\mathbb{N}}$, $\mathcal{S}_s^x \cap \mathcal{S}_t^{x'} \neq \emptyset$ only when $|s - t| \leq 1$. Fig. 1 gives the geometric explanation of the result.

For any $r \in \mathcal{R}$, assuming $r \in \mathcal{S}_i^x \cap \mathcal{S}_j^{x'}$, there must be $j \in \{i-1, i, i+1\}$ by the result in the above paragraph. Therefore,

$$\frac{p^x(r)}{p^{x'}(r)} = \frac{e^{-i\epsilon}}{e^{-j\epsilon}} \times \frac{\alpha^{x'}}{\alpha^x} \leq e^\epsilon \times e^\epsilon = e^{2\epsilon}. \quad (6)$$

The density of the Staircase mechanism is shown in Fig. 2(a).

The proof is complete.

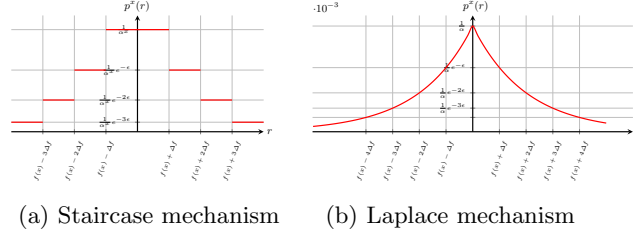


Fig. 2: The density functions of the Staircase mechanism and the Laplace mechanism

The Staircase mechanism embodies the principle of sensitivity-based mechanisms. That is, sensitivity-based mechanisms assign more high probability for those points $r \in \mathcal{R}$ that is more near to $f(x)$ when the input dataset is x . Intuitively, the assignments are reasonable since, for a point $r \in \mathbb{R}$, the distance $d(r, f(x))$ denotes the distortion extent when inputting x but outputting r . However, since the Staircase mechanism is greatly determined by the global sensitivity Δf , it works bad when the global sensitivity Δf is large. We now present a different mechanism. This mechanism is based on the observation that the Staircase mechanism is mainly determined by the metric d but with seldom relation with the metric \bar{d} , i.e., the metric for privacy. Therefore, we design a mechanism which is mainly based on the metric \bar{d} . Since \bar{d} is the metric for privacy as discussed in Section 3.6, the idea of the mechanism would be *first achieving privacy and then improving utility*.

Theorem 2. *For the dataset metric space (\mathcal{D}, \bar{d}) and a query function f , let (\mathcal{R}, d) be the value metric space of f . Set*

$$\mathcal{I}_0^x = \{f(x)\} \text{ and } \mathcal{I}_i^x = \cup_{y \in \mathcal{N}_i^x} \mathcal{I}_0^y - \mathcal{A}_{i-1}^x, \quad x \in \mathcal{D}, i \in \mathbb{N} \quad (7)$$

where $\mathcal{A}_{i-1}^x = \cup_{y \in \bar{\mathcal{N}}_{i-1}^x} \mathcal{I}_0^y$ and where $\mathcal{N}_i^x, \bar{\mathcal{N}}_i^x$ are defined as in Section 3.1. For the mechanism \mathcal{M} and a dataset $x \in \mathcal{D}$, let the probability distribution $p^x(r)$ of $\mathcal{M}(x)$ be

$$p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}, \quad \text{when } r \in \mathcal{I}_i^x,$$

where $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{I}_i^x)$. Then the mechanism \mathcal{M} satisfies 2ϵ -differential privacy.

Before proving the theorem, we first explain the idea of the mechanism. Being contrary to partition the codomain \mathcal{R} by the metric d as in the Staircase

mechanism, we partition it by the metric \bar{d} . Specifically, we first partition the dataset universe \mathcal{D} according to the metric \bar{d} , which generates the sequences $\{\mathcal{N}_i^x : i \in \bar{\mathbb{N}}\}$. Then, by using the mapping f , we can map the partition onto the codomain \mathcal{R} to partition \mathcal{R} accordingly, which generates the sequences $\{\mathcal{I}_i^x : i \in \bar{\mathbb{N}}\}$. That is, roughly speaking, each \mathcal{I}_i^x contains all of the values of the datasets in \mathcal{N}_i^x . The density $p^x(r)$ of $\mathcal{M}(x)$ is assigned as: the less of i , the more high probability to those points in \mathcal{I}_i^x .

The construction of the mechanism in Theorem 2 uses the same technique with the Staircase mechanism in Theorem 1: they both partition the value metric space \mathcal{R} into set sequences. However, the later uses the metric d to partition the set \mathcal{R} but the former uses the metric \bar{d} . The former mechanism can be explained as *first considering utility and then measuring privacy*, but the later mechanism can be explained as *first achieving privacy and then measuring utility*.

We now give a simple example to explain how to obtain $\{\mathcal{I}_i^x : i \in \bar{\mathbb{N}}\}$.

Example 7. Let the function $f : \mathbb{R} \rightarrow \mathbb{R}$ be defined as

$$f(x) = \begin{cases} x & \text{if } x \text{ is a rational number} \\ -x & \text{if } x \text{ is an irrational number} \end{cases} \quad (8)$$

Let d denote the Euclidean distance on \mathbb{R} . Then f is a mapping between the dataset metric space (d, \mathbb{R}) and the value metric space (d, \mathbb{R}) . We denote the first metric space as (\bar{d}, \mathcal{D}) , and the second as (d, \mathcal{R}) for the convenience of illustration.

For the function f and one $x \in \mathcal{D}$ in example 7, we have

$$\mathcal{N}_i^x = \{y \in \mathcal{D} : i - 1 < |x - y| \leq i\}$$

and

$$\begin{aligned} \mathcal{I}_i^x &= \cup_{y \in \mathcal{N}_i^x} \mathcal{I}_0^y - \mathcal{A}_{i-1}^x \\ &= [x - i, x - (i - 1))_{\mathbb{Q}} \cup (x + i - 1, x + i]_{\mathbb{Q}} \cup [-x - i, -x - (i - 1))_{\bar{\mathbb{Q}}} \cup (-x + i - 1, -x + i]_{\bar{\mathbb{Q}}}, \end{aligned}$$

where $[a, b]_{\mathbb{Q}} = [a, b] \cap \mathbb{Q}$, $[a, b]_{\bar{\mathbb{Q}}} = [a, b] \cap \bar{\mathbb{Q}}$, and $\bar{\mathbb{Q}} = \mathbb{R} - \mathbb{Q}$. Note that the set \mathcal{A}_{i-1}^x is used to delete those points appeared both in \mathcal{I}_i^x and $\cup_{j=0}^{i-1} \mathcal{I}_j^x$ from \mathcal{I}_i^x . Figure 3 shows the geometric explanation of \mathcal{N}_i^x and \mathcal{I}_i^x . From Figure 3, we can find that $\{\mathcal{I}_i^x, i \in \bar{\mathbb{N}}\}$ is a partition of the r -axis, i.e., the codomain \mathcal{R} of f . Let $x, x' \in \mathcal{D}$ be two neighbors. From Figure 3, we can also verify that, for any $s, t \in \bar{\mathbb{N}}$ such that $|s - t| \geq 2$, there is $\mathcal{I}_t^x \cap \mathcal{I}_s^{x'} = \emptyset$.

In general, we have the following lemma.

Lemma 1. *The set sequence $\{\mathcal{I}_i^x : i \in \bar{\mathbb{N}}\}$ in Theorem 2 is a partition of \mathcal{R} . That is, $\cup_{i \in \bar{\mathbb{N}}} \mathcal{I}_i^x = \mathcal{R}$ and, for any $i \neq j$, there is $\mathcal{I}_i^x \cap \mathcal{I}_j^x = \emptyset$. Moreover, letting $x, x' \in \mathcal{D}$ be neighbors, then, for any $s, t \in \bar{\mathbb{N}}$ such that $|s - t| \geq 2$, there is $\mathcal{I}_t^x \cap \mathcal{I}_s^{x'} = \emptyset$.*

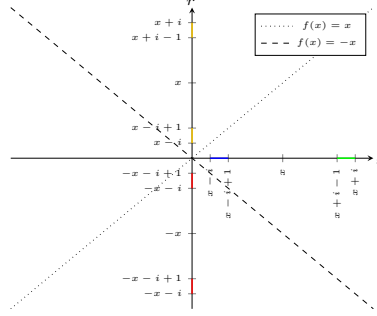


Fig. 3: The set \mathcal{N}_i^x is shown as the blue and the green lines in x -axis. The set \mathcal{I}_i^x is shown as the rational points in the yellow lines and the irrational points in the red lines in r -axis.

Proof. We first prove $\cup_{i=0}^t \mathcal{I}_i^x = \mathcal{A}_t^x$ for all $t \in \bar{\mathbb{N}}$. Note that $\mathcal{I}_i^x \subseteq \cup_{y \in \mathcal{N}_i^x} \mathcal{I}_0^y \subseteq \mathcal{A}_t^x$ for all $i \leq t$. We have $\cup_{i=0}^t \mathcal{I}_i^x \subseteq \mathcal{A}_t^x$ for all $t \in \bar{\mathbb{N}}$. We next prove $\cup_{i=0}^t \mathcal{I}_i^x \supseteq \mathcal{A}_t^x$ by induction. Obviously, $\cup_{i=0}^0 \mathcal{I}_i^x = \mathcal{A}_0^x$; Assume that $\cup_{i=0}^t \mathcal{I}_i^x \supseteq \mathcal{A}_t^x$ for all $t < k$; For any $r \in \mathcal{A}_k^x$, there exist $y \in \mathcal{N}_k^x$ and a minimum $j \in \bar{\mathbb{N}}$ such that $r \in \mathcal{A}_j^x$ and $\bar{d}(x, y) = j \leq k$. If $j < k$ we have $r \in \mathcal{A}_j^x \subseteq \cup_{i=0}^j \mathcal{I}_i^x \subseteq \cup_{i=0}^k \mathcal{I}_i^x$ by the assumption. If $j = k$ then there is no $i < k$ such that $r \in \mathcal{A}_i^x$ by the minimality of j . We have $r \in \mathcal{A}_k^x - \mathcal{A}_{k-1}^x = \cup_{y \in \mathcal{N}_k^x} \mathcal{I}_0^y - \mathcal{A}_{k-1}^x = \mathcal{I}_k^x \subseteq \cup_{i=0}^k \mathcal{I}_i^x$. In conclusion, $\cup_{i=0}^t \mathcal{I}_i^x = \mathcal{A}_t^x$ for all $x \in \mathcal{D}$, $t \in \bar{\mathbb{N}}$.

Note that $\cup_{i \in \bar{\mathbb{N}}} \mathcal{I}_i^x = \mathcal{R}$ is a direct corollary of $\cup_{i=0}^t \mathcal{I}_i^x = \mathcal{A}_t^x$ for all $x \in \mathcal{D}$, $t \in \bar{\mathbb{N}}$. We next prove $\mathcal{I}_t^x \cap \mathcal{I}_s^{x'} = \emptyset$. Without loss of generality, set $s \leq t-2$. Since $\mathcal{I}_t^x = \cup_{y \in \mathcal{N}_t^x} \mathcal{I}_0^y - \mathcal{A}_{t-1}^x$, for any $r \in \mathcal{I}_t^x$, we have that there exists $y \in \mathcal{D}$ such that $\bar{d}(x, y) = t$ and $r = f(y)$, and that for any $y' \in \mathcal{D}$ such that $\bar{d}(x, y') \leq t-1$ there has $r \neq f(y')$. On the other hand, for any $r' \in \mathcal{I}_s^{x'}$, there exists $\hat{x} \in \mathcal{D}$ such that $\bar{d}(\hat{x}, x') = s$ and $r' = f(\hat{x})$. Since $\bar{d}(x, \hat{x}) \leq \bar{d}(x, x') + \bar{d}(x', \hat{x}) = 1 + s \leq t-1$, we have $r \neq r'$, which implies $\mathcal{I}_t^x \cap \mathcal{I}_s^{x'} = \emptyset$.

The claims are proved.

Proof (The proof of Theorem 2). Let $x, x' \in \mathcal{D}$ be neighbors. For any $r \in \mathcal{R}$, assuming $r \in \mathcal{I}_i^x \cap \mathcal{I}_j^{x'}$, there is $j \in \{i-1, i, i+1\}$ by Lemma 1. Therefore,

$$\frac{p^x(r)}{p^{x'}(r)} = \frac{e^{-i\epsilon}}{e^{-j\epsilon}} \times \frac{\alpha^{x'}}{\alpha^x} \leq e^\epsilon \times e^\epsilon = e^{2\epsilon} \quad (9)$$

The proof is complete.

We now use the above two mechanisms to explain the mechanisms in related works.

Example 8 (Global sensitivity mechanism). Let $q^x(r)$ be the probability distribution of the equation (4) in [1] and let $p^x(r)$ be the probability distribution of

the Staircase mechanisms in Theorem 1. Then

$$q^x(r) = \frac{1}{\beta^x} e^{-\frac{d(r, f(x))}{\Delta f} \epsilon}. \quad (10)$$

Since $(i-1)\Delta f < d(f(x), r) \leq i\Delta f$ if and only if $r \in \mathcal{S}_i^x$, therefore,

$$q^x(r) = \frac{1}{\beta^x} e^{-b\epsilon} \text{ if and only if } p^x(r) = \frac{1}{\alpha^x} e^{-\lceil b \rceil \epsilon}, \quad (11)$$

where both of α^x and β^x are normalizations. Hence, the Staircase mechanism in Theorem 1 can be seen as the discrete variant of the global sensitivity mechanism in [1].

Example 9 (Staircase mechanism). The mechanism in [2,3] can be considered as a general form of the Staircase mechanism in Theorem 1. The only difference is to set the set sequence $\{\mathcal{S}_i^x, i \in \bar{\mathbb{N}}\}$ as

$$\mathcal{S}_i^x = \{r \in \mathcal{R} : (i-1)\Delta f + \gamma < d(f(x), r) \leq i\Delta f + \gamma\}, \quad i \in \bar{\mathbb{N}},$$

where $0 \leq \gamma < \Delta$ is constant.

Example 10 (Ladder mechanism). The mechanism in [4], which we called *Ladder mechanism*, is a local sensitivity-based mechanism. It can be considered as a variant of the Staircase mechanism in Theorem 1. The only difference is to set the set sequence $\{\mathcal{S}_i^x, i \in \bar{\mathbb{N}}\}$ as

$$\mathcal{S}_i^x = \left\{ r \in \mathcal{R} : \sum_{j=0}^{i-1} \Delta_j^x < d(f(x), r) \leq \sum_{j=0}^i \Delta_j^x \right\}, \quad i \in \bar{\mathbb{N}}.$$

The reasons are as follows. Let $x, x' \in \mathcal{D}$ be two neighboring datasets. For any $y \in \mathcal{D}$, if $\bar{d}(x', y) \leq i$, then $\bar{d}(x, y) \leq \bar{d}(x, x') + \bar{d}(x', y) \leq 1 + i$. We have

$$\{y \in \mathcal{D} : \bar{d}(x', y) \leq i\} \subseteq \{y \in \mathcal{D} : \bar{d}(x, y) \leq i+1\}.$$

Therefore,

$$\Delta_i^{x'} = \max_{y \in \mathcal{D} : \bar{d}(x', y) \leq i} \Delta_0^y \leq \max_{y \in \mathcal{D} : \bar{d}(x, y) \leq i+1} \Delta_0^y = \Delta_{i+1}^x.$$

Furthermore, since $f(x') \in \{r \in \mathcal{R} : d(f(x), r) \leq \Delta_0^x\}$, we have that $\{\mathcal{S}_i^x : i \in \bar{\mathbb{N}}\}$ and $\{\mathcal{S}_i^{x'} : i \in \bar{\mathbb{N}}\}$ satisfy that, for any two integers $s, t \in \bar{\mathbb{N}}$, $\mathcal{S}_s^x \cap \mathcal{S}_t^{x'} \neq \emptyset$ only when $|s - t| \leq 1$. Fig. 4 shows the geometric explanation. Therefore, the above mechanism satisfies 2ϵ -differential privacy.

Example 11 (K-norm mechanism [5,6]). The K -norm mechanism can be explained as a continuous variant of a special case of the mechanism in Theorem 2. To see this, we need to first represent the K -norm mechanism as the language of this paper. Set $\mathcal{D} = \mathbb{R}^n$, $\mathcal{R} = \mathcal{R}^d$ and the linear query F . In [5,6], two

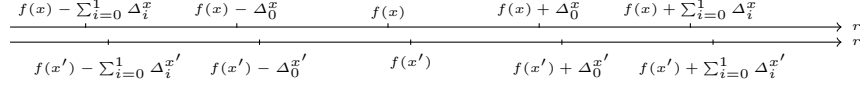


Fig. 4: Partition of \mathcal{R} in Ladder mechanism, where the upper, below lines show the set sequences $\{\mathcal{S}_i^x : i \in \mathbb{N}\}$, $\{\mathcal{S}_i^{x'} : i \in \mathbb{N}\}$, respectively

datasets $x, x' \in \mathcal{D}$ are said to be neighbors if $\|x - x'\|_1 \leq 1$. For any $x \in \mathcal{D}$, set $\mathcal{N}_i^x = \{y \in \mathcal{D} : i - 1 < \|x - y\|_1 \leq i\}$. We have $\mathcal{I}_i^x = \{F(y) : y \in \mathcal{N}_i^x\}$. The ℓ_1 unit ball $B_1^n = \mathcal{N}^x - x$ and the set $K = FB_1^n = \mathcal{A}_1^x - F(x)$. For any real number $b > 0$, one can verify that

$$\mathcal{A}_{[b]}^x - F(x) = [b]FB_1^n = [b]K \subseteq bK \subseteq [b]K = [b]FB_1^n = \mathcal{A}_{[b]}^x - F(x). \quad (12)$$

Therefore, for any $r \in \mathcal{R}$, if $\|F(x) - r\|_K = b$, then $r \in \mathcal{I}_{[b]}^x = \mathcal{A}_{[b]}^x - \mathcal{A}_{[b]}^x$. Set $q^x(r)$ be the probability distribution of the K -norm mechanism when the dataset is x , we have

$$q^x(r) = \frac{1}{\beta^x} e^{-b\epsilon} \text{ if and only if } p^x(r) = \frac{1}{\alpha^x} e^{-[b]\epsilon}, \quad (13)$$

where β^x is the normalization. Therefore, the K -norm mechanism can be explained as the discrete case of a special case of the mechanism in Theorem 2.

4.2 The Compositional Mechanisms

We now present two mechanisms which can be considered as compositions of the mechanisms in Theorem 1 and Theorem 2. The mechanisms mentioned above are based either on the first metric \bar{d} or on the second metric d but not on both. In this section, we design compositional mechanisms which are based on both of the two metrics \bar{d}, d .

We first give a variant of the mechanism in Theorem 2, which incorporates the second metric d into the mechanism design. For each $x \in \mathcal{D}$, equip it a nonnegative real number δ^x . Set $\mathcal{R}_0^{x, \delta^x} = \{r \in \mathcal{R} : d(f(x), r) \leq \delta^x\}$. For $i \in \mathbb{N}$, set $\mathcal{R}_i^{x, \delta^x} = \cup_{y \in \mathcal{N}_i^x} \mathcal{R}_0^{y, \delta^y} - \mathcal{B}_{i-1}^{x, \delta}$, where $\mathcal{B}_{i-1}^{x, \delta} = \cup_{y \in \mathcal{N}_{i-1}^x} \mathcal{R}_0^{y, \delta^y}$. In the following, we denote $\mathcal{R}_i^x = \mathcal{R}_i^{x, \delta^x}$ and $\mathcal{B}_i^x = \mathcal{B}_i^{x, \delta}$ for notational simplicity when there is no ambiguity. We construct a mechanism \mathcal{M} as follows.

Theorem 3. *For any $x \in \mathcal{D}$ and any $r \in \mathcal{R}$ set the density function of $\mathcal{M}(x)$ as $p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}$ if $r \in \mathcal{R}_i^x$, where the normalizer $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x)$. Then \mathcal{M} is 2ϵ -differentially private.*

Before proving Theorem 3, we give some interpretations about Theorem 3 and the set sequence $\{\mathcal{R}_i^x : i \in \mathbb{N}\}$. The construction of the set sequence $\{\mathcal{R}_i^x : i \in \mathbb{N}\}$ is similar with the one of the set sequence $\{\mathcal{I}_i^x : i \in \mathbb{N}\}$. Their main difference is

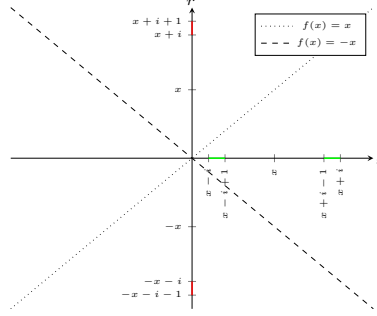


Fig. 5: The set \mathcal{N}_i^x is shown as the green lines in x -axis. The set \mathcal{R}_i^x (for $i \geq x$) is shown as the red lines in r -axis.

to substitute a ball $\mathcal{R}_0^x = \{r \in \mathcal{R} : d(r, f(x)) \leq \delta^x\}$ for the singleton $\mathcal{I}_0^x = \{f(x)\}$ from the construction of \mathcal{I}_i^x . This substitution will assign the points in \mathcal{R} near to $f(x)$ the highest probabilities, the aim of which is obvious: improve the utility. In this way, we want to remedy the disadvantage of the construction of $\{\mathcal{I}_i^x : i \in \bar{\mathbb{N}}\}$, which only assigns high probabilities to those $f(y) \in \mathcal{R}$ if $\bar{d}(x, y)$ being small. The role of the set \mathcal{B}_{i-1}^x is similar with the role of the set \mathcal{A}_{i-1}^x in $\{\mathcal{I}_i^x : i \in \bar{\mathbb{N}}\}$. That is, it is used to delete the overlapped elements of the sets \mathcal{R}_i^x and $\cap_{j=0}^{i-1} \mathcal{R}_j^x$ from the set \mathcal{R}_i^x . It thus appears that the construction of the mechanism in Theorem 3 employs both of the metrics \bar{d}, d .

Next, we present a concrete example about how to construct the set sequence $\{\mathcal{R}_i^x : i \in \bar{\mathbb{N}}\}$ by using Example 7. In Example 7, if $x \in \mathbb{Z}_+$, then $\mathcal{R}_0^x = [x-1, x+1]$ and

$$\begin{aligned} \mathcal{R}_1^x &= \cup_{y \in \mathcal{N}_1^x} \mathcal{R}_0^y - \mathcal{R}_0^x = \cup_{y \in \mathcal{N}_1^x \cap \mathbb{Q}} \mathcal{R}_0^y \cup_{y \in \mathcal{N}_1^x \cap \bar{\mathbb{Q}}} \mathcal{R}_0^y - \mathcal{R}_0^x \\ &= [-x-2, -x+2] \cup [x-2, x-1] \cup (x+1, x+2]. \end{aligned}$$

For $2 \leq i \leq x-1$, we have

$$\begin{aligned} \mathcal{R}_i^x &= \cup_{y \in \mathcal{N}_i^x} \mathcal{R}_0^y - \cup_{j=0}^{i-1} \mathcal{R}_j^x \\ &= [-x-(i+1), -x-i] \cup (-x+i, -x+i+1] \cup [x-(i+1), x-i] \cup (x+i, x+i+1]. \end{aligned}$$

For $i \geq x$, we have

$$\mathcal{R}_i^x = [-x-(i+1), -x-i] \cup (x+i, x+i+1].$$

Fig. 5 shows the geometric explanation of \mathcal{N}_i^x and \mathcal{R}_i^x (for $i \geq x$). Other cases can be treated similarly. One can verify that, for any two neighbors $x, x' \in \mathcal{D}$, $\mathcal{R}_i^x \cap \mathcal{R}_s^{x'} \neq \emptyset$ only when $|t-s| \leq 1$. In general, we have the following lemma.

Lemma 2. *The set sequence $\{\mathcal{R}_i^x : i \in \bar{\mathbb{N}}\}$ in Theorem 3 is a partition of \mathcal{R} . That is, $\cup_{i \in \bar{\mathbb{N}}} \mathcal{R}_i^x = \mathcal{R}$ and, for any $i \neq j$, there is $\mathcal{R}_i^x \cap \mathcal{R}_j^x = \emptyset$. Moreover,*

letting $x, x' \in \mathcal{D}$ be neighbors, then, for any $s, t \in \bar{\mathbb{N}}$ such that $|s - t| \geq 2$, there is $\mathcal{R}_t^x \cap \mathcal{R}_s^{x'} = \emptyset$.

Proof. We first prove $\cup_{i=0}^t \mathcal{R}_i^x = \mathcal{B}_t^x$ for all $t \in \bar{\mathbb{N}}$. Note that $\mathcal{R}_i^x \subseteq \cup_{y \in \mathcal{N}_i^x} \mathcal{R}_0^y \subseteq \mathcal{B}_t^x$ for all $i \leq t$. We have $\cup_{i=0}^t \mathcal{R}_i^x \subseteq \mathcal{B}_t^x$ for all $t \in \bar{\mathbb{N}}$. We next prove $\cup_{i=0}^t \mathcal{R}_i^x \supseteq \mathcal{B}_t^x$ by induction. First, one can verify $\cup_{i=0}^0 \mathcal{R}_i^x = \mathcal{B}_0^x$; Second, assume that $\cup_{i=0}^t \mathcal{R}_i^x \supseteq \mathcal{B}_t^x$ for all $t < k$; Third, since for any $i < j$ there is $\mathcal{B}_i^x \subseteq \mathcal{B}_j^x$, therefore, for any $r \in \mathcal{B}_k^x$, there exists a minimum $j \in \bar{\mathbb{N}}$ satisfying $r \in \mathcal{B}_j^x$. If $j < k$, according to the above assumption, we have $r \in \mathcal{B}_j^x \subseteq \cup_{i=0}^j \mathcal{R}_i^x \subseteq \cup_{i=0}^k \mathcal{R}_i^x$. If $j = k$, there does not exist $i < k$ satisfying $r \in \mathcal{B}_i^x$ by the minimality of j . We then have $r \in \mathcal{B}_k^x - \mathcal{B}_{k-1}^x = \cup_{y \in \mathcal{N}_k^x} \mathcal{R}_0^y - \mathcal{B}_{k-1}^x = \mathcal{R}_k^x \subseteq \cup_{i=0}^k \mathcal{R}_i^x$. Therefore, for any $x \in \mathcal{D}$ and any $t \in \bar{\mathbb{N}}$, there is $\cup_{i=0}^t \mathcal{R}_i^x = \mathcal{B}_t^x$. Moreover, since $\cup_{i=0}^t \mathcal{R}_i^x = \mathcal{B}_t^x$, $x \in \mathcal{D}$, $t \in \bar{\mathbb{N}}$, we have $\cup_{i \in \bar{\mathbb{N}}} \mathcal{R}_i^x = \mathcal{R}$.

Next, we prove $\mathcal{R}_i^x \cap \mathcal{R}_j^x = \emptyset$ for any $i \neq j$. Without loss of generality, set $i < j$. Since $\mathcal{R}_j^x = \cup_{y \in \mathcal{N}_j^x} \mathcal{R}_0^y - \mathcal{B}_{j-1}^x$, then $\mathcal{R}_j^x \cap \mathcal{B}_{j-1}^x = \emptyset$. According to $\cup_{i=0}^t \mathcal{R}_i^x = \mathcal{B}_t^x$, there is $\mathcal{B}_{j-1}^x \supseteq \mathcal{B}_i^x \supseteq \mathcal{R}_i^x$. Therefore, $\mathcal{R}_i^x \cap \mathcal{R}_j^x = \emptyset$.

Now we prove $\mathcal{R}_t^x \cap \mathcal{R}_s^{x'} = \emptyset$. Without loss of generality, set $s \leq t - 2$. Since $\mathcal{R}_t^x = \cup_{y \in \mathcal{N}_t^x} \mathcal{R}_0^y - \mathcal{B}_{t-1}^x$, therefore, for any $r \in \mathcal{R}_t^x$, there exists $y \in \mathcal{D}$ satisfying $\bar{d}(x, y) = t$, $r = f(y)$ and, for any $y' \in \mathcal{D}$ satisfying $\bar{d}(x, y') \leq t - 1$, there is $r \neq f(y')$. On the other hand, for any $r' \in \mathcal{R}_s^{x'}$, there exists $\hat{x} \in \mathcal{D}$ satisfying $\bar{d}(\hat{x}, x') = s$ and $r' = f(\hat{x})$. Since $\bar{d}(x, \hat{x}) \leq \bar{d}(x, x') + \bar{d}(x', \hat{x}) = 1 + s \leq t - 1$, we have $r \neq r'$. Therefore, $\mathcal{R}_t^x \cap \mathcal{R}_s^{x'} = \emptyset$.

The claim is proved.

Proof (The proof of Theorem 3). The proof is similar with the proof of Theorem 2. The difference is just substituting Lemma 2 for Lemma 1.

We now give a variant of the Staircase mechanism in Theorem 1, which incorporates the first metric \bar{d} into the mechanism design. The details are as follows. For any $r \in \mathcal{R}$, set

$$f^{-1}(r) = \{z \in \mathcal{D} : f(z) = r\}. \quad (14)$$

Then $f^{-1}(f(x)) = \{z \in \mathcal{D} : f(z) = f(x)\}$. For each $r \in \mathcal{R}$, equip it a real number $\delta^r \geq 0$. For each $x \in \mathcal{D}$, set

$$\mathcal{T}_0^{x, \delta} = \cup_{z \in f^{-1}(f(x))} \{f(y) \in \mathcal{R} : y \in \mathcal{D} \wedge \bar{d}(z, y) \leq \delta^r \wedge f(x) = r\}. \quad (15)$$

One can verify that, for any $x, y \in \mathcal{D}$, if $f(x) = f(y)$, there is

$$\mathcal{T}_0^{x, \delta} = \mathcal{T}_0^{y, \delta}. \quad (16)$$

For each $i \in \mathbb{N}$, set

$$\mathcal{T}_i^{x, \delta} = \cup_{y \in \mathcal{D} : f(y) \in \mathcal{S}_i^x} \mathcal{T}_0^{y, \delta} - \mathcal{C}_{i-1}^{x, \delta}, \quad (17)$$

where $\mathcal{C}_{i-1}^{x, \delta} = \cup_{y \in \mathcal{D} : f(y) \in \bar{\mathcal{S}}_{i-1}^x} \mathcal{T}_0^{y, \delta}$ and $\bar{\mathcal{S}}_i^x = \cup_{i=0}^i \mathcal{S}_i^x$. In the following, when there is no ambiguity, set $\mathcal{T}_i^x = \mathcal{T}_i^{x, \delta}$ and $\mathcal{C}_i^x = \mathcal{C}_i^{x, \delta}$.

We give an explanation about the set sequence $\{\mathcal{T}_i^x : i \in \bar{\mathbb{N}}\}$. For simplicity, we assume that f is injective. The idea of the construction of $\{\mathcal{T}_i^x : i \in \bar{\mathbb{N}}\}$ is similar with $\{\mathcal{S}_i^x : i \in \bar{\mathbb{N}}\}$. The main difference is to substitute a ball $\mathcal{T}_0^{x,\delta} = \{f(y) \in \mathcal{R} : y \in \mathcal{D} \wedge \bar{d}(y, x) \leq \delta^r \wedge f(x) = r\}$ for the singleton $\mathcal{S}_0^x = \{f(x)\}$. The set $\mathcal{C}_{i-1}^{x,\delta}$ is used to delete the overlapped elements of \mathcal{T}_i^x and $\cap_{j=0}^{i-1} \mathcal{T}_j^x$ from \mathcal{T}_i^x . Similar to Theorem 1, we have the following theorem.

Theorem 4. *For the mechanism \mathcal{M} and a dataset $x \in \mathcal{D}$, let the probability distribution $p^x(r)$ of $\mathcal{M}(x)$ be*

$$p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}, \quad \text{when } r \in \mathcal{T}_i^x,$$

where $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{T}_i^x)$. Then the mechanism \mathcal{M} satisfies 2ϵ -differential privacy.

Proof. The proof of Theorem 4 is very similar with the one of Theorem 3 and therefore is omitted.

4.3 The Third Category Mechanisms

We now construct two mechanisms, each of which is a variant of Theorem 4 or Theorem 3 respectively.

Theorem 5. *Set $\mathcal{E} \subset \mathcal{R}$ and $t \in \mathbb{R}_+$. Set $\mathcal{E}_i^x = \mathcal{T}_i^x \cap \mathcal{E}$ and $\bar{\mathcal{E}}_i^x = \mathcal{T}_i^x \cap \bar{\mathcal{E}}$, where $\bar{\mathcal{E}} = \mathcal{R} - \mathcal{E}$. When $r \in \mathcal{E}_i^x$, set $p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}$; when $r \in \bar{\mathcal{E}}_i^x$, set $p^x(r) = \frac{1}{\alpha^x} e^{-(i+t)\epsilon}$, where $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{E}_i^x) + \sum_{i=0}^{\infty} e^{-(i+t)\epsilon} \mu(\bar{\mathcal{E}}_i^x)$ is the normalization. Then $p^x(r)$ satisfies 2ϵ -differential privacy.*

Theorem 6. *Set $\mathcal{E} \subset \mathcal{R}$ and $t \in \mathbb{R}_+$. Set $\mathcal{E}_i^x = \mathcal{R}_i^x \cap \mathcal{E}$ and $\bar{\mathcal{E}}_i^x = \mathcal{R}_i^x \cap \bar{\mathcal{E}}$, where $\bar{\mathcal{E}} = \mathcal{R} - \mathcal{E}$. When $r \in \mathcal{E}_i^x$, set $p^x(r) = \frac{1}{\alpha^x} e^{-i\epsilon}$; when $r \in \bar{\mathcal{E}}_i^x$, set $p^x(r) = \frac{1}{\alpha^x} e^{-(i+t)\epsilon}$, where $\alpha^x = \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{E}_i^x) + \sum_{i=0}^{\infty} e^{-(i+t)\epsilon} \mu(\bar{\mathcal{E}}_i^x)$ is the normalization. Then $p^x(r)$ satisfies 2ϵ -differential privacy.*

The proof of the above two theorem are the same with Theorem 4 or Theorem 3 and are omitted.

We now explain the above two mechanisms. Let $\mathcal{S} \subset \mathcal{D}$ and let $\mathcal{E} = \{f(x) : x \in \mathcal{S}\}$. The parameter t is used to balance the utility of the mechanism \mathcal{M} between the datasets in \mathcal{S} and the datasets in $\bar{\mathcal{S}} = \mathcal{D} - \mathcal{S}$, where the set \mathcal{S} contains those datasets of importance, such as, each datasets in \mathcal{S} has high appearing probability. The idea of the above two mechanisms is: The outputs of the datasets in \mathcal{S} should obtain high utility than those in $\bar{\mathcal{S}}$. The coefficient $e^{-t\epsilon}$ is used as the penalty factor of the utility of those datasets in $\bar{\mathcal{S}}$. The mechanisms try to improve the overall utility of the mechanism by tuning the parameter t . In the mechanisms, when $\mathcal{E} = \mathcal{R}$, they changes to the one in Theorem 4 or Theorem 3.

4.4 Discussion

As discussed in 3.6, the tradeoff between privacy and utility can be achieved by designing randomized mappings between two metric spaces. In Section 4 we present three categories of mechanisms to implement the tradeoff.

In Section 4.1 we present two basic mechanisms. The Staircase mechanism in Theorem 1 can be considered as a prototype of the sensitivity-based mechanisms. In technical aspect, it mainly uses the metric d of the value metric space (\mathcal{R}, d) to construct mechanisms. On the contrary, the mechanism in Theorem 2 only uses the metric \bar{d} of the dataset metric space (\mathcal{D}, \bar{d}) to construct mechanisms. Since d, \bar{d} are metrics to measure the utility, the privacy respectively as noted in 3.6, we can explain the Staircase mechanism and the mechanism in Theorem 2 as *utility oriented* and *privacy oriented*, respectively.

In Section 4.2 we present another two composition mechanisms, each of which use both of the metrics d, \bar{d} . The mechanism in Theorem 4 can be considered as *first keeping utility then measuring privacy*. On the contrary, the mechanism in Theorem 3 can be considered as *first achieving privacy then improving utility*.

In Section 4.3 we also present two mechanisms. The two mechanisms aim to balance the utility of a mechanism among different datasets.

One needing to be emphasized is that the proofs of the mechanisms in Theorem 2, in Theorem 3 and in Theorem 6 do not use the symmetric property and the triangular inequality property of the metric d . This implies that we can substitute a distortion function $L^x(r)$, which does not have the above two properties, for the metric $d(f(x), r)$.

From the above discussion we can find that the mechanism designing techniques used in Section 4 are powerful. Furthermore, they are *universal methods* as the sensitivity-based methods, which are applicable for all queries in Section 3.

5 Parameter Tuning and Utility Analysis

In this section, we consider the parameter tuning of the mechanisms in Section 4.2. The mechanisms in Section 4.2 are compositional mechanisms, which combine both of the two metrics \bar{d}, d to design mechanisms, and in which there are parameters $\{\delta^r, r \in \mathcal{R}\}, \{\delta^x, x \in \mathcal{D}\}$. The aims of the compositional mechanisms are to improve the utility of the mechanisms by tuning the parameters. We only analyze the mechanism in Theorem 3. The mechanism in Theorem 4 can be treated similarly.

We first consider the relation of P_T^x with the parameters $\{\delta^x, x \in \mathcal{D}\}$ in Theorem 3, where P_T^x can be evaluated by the equation

$$P_T^x = \Pr[\mathcal{M}(x) \in C_T^x] = \frac{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap C_T^x)}{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x)}, \quad (18)$$

with $C_T^x = \{r \in \mathcal{R} : d(f(x), r) \leq T\}$. Obviously, when $\delta^x = 0, x \in \mathcal{D}$, the above value P_T^x is the same as the corresponding value of the mechanism in Theorem

2. Set $\bar{C}_T^x = \mathcal{R} - C_T^x$, we have

$$\sum_{i=0}^{\infty} \mu(\mathcal{R}_i^x \cap \bar{C}_T^x) = \mu(\bar{C}_T^x), \quad \sum_{i=0}^{\infty} \mu(\mathcal{R}_i^x \cap C_T^x) = \mu(C_T^x), \quad (19)$$

$$\sum_{i=0}^{\infty} \mu(\mathcal{R}_i^x) = \mu(\mathcal{R}), \quad \mu(\mathcal{R}_i^x) = \mu(\mathcal{R}_i^x \cap C_T^x) + \mu(\mathcal{R}_i^x \cap \bar{C}_T^x). \quad (20)$$

Then

$$P_T^x = \frac{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap C_T^x)}{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap C_T^x) + \sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap \bar{C}_T^x)} \quad (21)$$

$$= \frac{1}{1 + \frac{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap \bar{C}_T^x)}{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap C_T^x)}} \quad (22)$$

Therefore, in order to raise P_T^x , we should decrease the value of $\frac{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap \bar{C}_T^x)}{\sum_{i=0}^{\infty} e^{-i\epsilon} \mu(\mathcal{R}_i^x \cap C_T^x)}$.

The later can be achieved by decreasing the value $\frac{\mu(\mathcal{R}_i^x \cap \bar{C}_T^x)}{\mu(\mathcal{R}_i^x \cap C_T^x)}$ for smaller i . One needs to be mentioned is that each dataset $x \in \mathcal{D}$ has its own δ^x . The change of δ^x of any dataset x will have effect on all the P_T^y s of all other datasets $y \in \mathcal{D} \setminus \{x\}$. That is, for a dataset $x \in \mathcal{D}$, raising its δ^x will raise $\mu(\mathcal{R}_0^x)$ and then may raise P_T^x . However, this is based on prerequisite of the unchanging of all other datasets' δ^x s. If all the δ^x s of all the datasets raise, the variation of P_T^x s would be difficult to be determined. By the above analysis, we could find that this is a complicated balance problem, not just balance between utility and privacy, but also balance among the utilities of different datasets.

We next analyze the mechanisms in Theorem 3 by the expected distortion $\mathbb{E}[d(r, f(x))]$. The utility of the dataset $x \in \mathcal{D}$ is

$$\mathbb{E}[d(r, f(x))] = \int_{r \in \mathcal{R}} d(f(x), r) p^x(r) \mu(dr), \quad (23)$$

where $p^x(r)$ is the probability distribution of $\mathcal{M}(x)$. By inputting the probability distribution of the mechanism in Theorem 3, we have

$$\mathbb{E}[d(r, f(x))] = \frac{\sum_{i=0}^{\infty} \exp(-i\epsilon) \int_{r \in \mathcal{R}_i^x} d(f(x), r) \mu(dr)}{\sum_{i=0}^{\infty} \exp(-i\epsilon) \int_{r \in \mathcal{R}_i^x} \mu(dr)}, \quad (24)$$

where

$$\sum_{i=0}^{\infty} \int_{r \in \mathcal{R}_i^x} d(f(x), r) \mu(dr) = \int_{r \in \mathcal{R}} d(f(x), r) \mu(dr), \quad \sum_{i=0}^{\infty} \int_{r \in \mathcal{R}_i^x} \mu(dr) = \int_{r \in \mathcal{R}} \mu(dr) = \mu(\mathcal{R}).$$

Setting $a_i^x = \int_{r \in \mathcal{R}_i^x} d(f(x), r) \mu(dr)$, $b_i^x = \mu(\mathcal{R}_i^x)$, $a^x = \int_{r \in \mathcal{R}} d(f(x), r) \mu(dr)$, $b = \mu(\mathcal{R})$, there is

$$\mathbb{E}[d(r, f(x))] = \frac{\sum_{i=0}^{\infty} \exp(-i\epsilon) a_i^x}{\sum_{i=0}^{\infty} \exp(-i\epsilon) b_i^x}. \quad (25)$$

In order to simplify the analysis, we next assume that all the datasets have the same δ , i.e., $\delta^x = \delta, \forall x \in \mathcal{D}$. Then a_i^x, b_i^x can be seen as functions of δ and the same as $\mathbb{E}[d(r, f(x))]$. The derivative of $\mathbb{E}[d(r, f(x))]$ about δ is

$$\frac{d\mathbb{E}[d(r, f(x))]}{d\delta} = \frac{\sum_{j=0}^{\infty} \sum_{i=0}^{\infty} \exp(-i\epsilon) \exp(-j\epsilon) (\frac{da_i^x}{d\delta} b_j^x - \frac{db_i^x}{d\delta} a_j^x)}{(\sum_{i=0}^{\infty} \exp(-i\epsilon) b_i^x)^2}. \quad (26)$$

When δ raises, the variation of b_i^x is not determined since b_i^x is the volume of \mathcal{R}_i^x . According to the definition of \mathcal{R}_0^x , $\frac{db_0^x}{d\delta} \geq 0$, $\frac{da_0^x}{d\delta} \geq 0$. Since $a^x = \sum_{i=0}^{\infty} a_i^x$ and $b = \sum_{i=0}^{\infty} b_i^x$, we have that the sign of each of the following quantities

$$\frac{da_i^x}{d\delta} b_j^x - \frac{db_i^x}{d\delta} a_j^x, i \in \bar{\mathbb{N}}, j \in \bar{\mathbb{N}} \quad (27)$$

is undetermined. Therefore, the sign of $\frac{d\mathbb{E}[d(r, f(x))]}{d\delta}$ is undermined. Hence, the variation of $\mathbb{E}[d(r, f(x))]$ is complicated and is hard to have a uniform conclusion for all query functions. We will analyze the quantity for some specific query functions by experiments in Section 6.

Discussion: In Section 5, we analyze the variation of the utilities of the mechanisms in Section 4.2. We find that the variation of the utilities of these mechanisms are very complicated when the parameters $\{\delta^r, r \in \mathcal{R}\}$, $\{\delta^x, x \in \mathcal{D}\}$ are changing. Therefore, we do not obtain any general and formal results about the variation currently, which will be one challenging future work. We will analyze the utilities for some specific query functions by experiments in Section 6.

We do not analyze the parameter tuning of the mechanisms in Section 4.3, which would be another future work.

6 Application

In this section, we apply the mechanism in Theorem 3 to the subgraph counting problem and the linear query problem. Recall that the mechanism in Theorem 2 is the special case of the one in Theorem 3, where $\delta^x = 0$ for all $x \in \mathcal{D}$.

6.1 Application in Subgraph Counting

Subgraph counting is one important problem in differential privacy [10,9,8,4]. We use the *edge differential privacy* as in [4]. That is, two graphs are said to be neighbors if the difference of their edges is 1. We count the number of *triangles* in a graph. The corresponding algorithm is shown in Algorithm 1, which implements the mechanism in Theorem 2 and outputs the set sequence $\{\mathcal{I}_k^{g_i} : k \in \bar{\mathbb{N}}\}$ of each graph $g_i \in \mathcal{D}$ in \mathcal{G}_n , where \mathcal{G}_n denotes the set of all the graphs with the number of nodes equals n and \mathcal{D} denotes the set of all non-isomorphic graphs in \mathcal{G}_n . The algorithm for the mechanism in Theorem 3 is similar with Algorithm 1 and is omitted.

Algorithm 1: Generate the set sequence $\{\mathcal{I}_k^{g_i}\}$ of each graph g_i

Input: The set of all non-isomorphic graphs $\mathcal{D} = \{g_1, g_2, \dots, g_m\}$ in \mathcal{G}_n
Output: The matrix I , where $I[k, i]$ is the set $\mathcal{I}_k^{g_i}$ of the graph g_i
 /* The matrix $adjG$ stores the neighboring relationship of graphs */
 1 **for** $i = 1$ **to** m **do**
 2 **for** $j = 1$ **to** m **do**
 3 **if** *If the graph g_j is a neighbor of distance k of g_i* **then**
 4 set $adjG[i, j] = k$
 /* The array tri stores the number of triangles of the graphs */
 5 **for** $i = 1$ **to** m **do**
 6 **if** *If the number of triangles in the graph g_i is k* **then**
 7 set $tri[i] = k$
 8 set $I[0, i] = \{k\}$
 /* The set $I[k, i]$ is the set $\mathcal{I}_k^{g_i}$ of the graph g_i */
 9 **for** $k = 1$ **to** $edgeNo$ **do** // $edgeNo = \frac{n(n-1)}{2}$
 10 **for** $i = 1$ **to** m **do**
 11 $Itemp = \text{NULL}$
 12 **for** $j = 1$ **to** m **do**
 13 **if** $adjG[i, j] == k$ **then**
 14 $Itemp = \text{union}(Itemp, tri[j])$ // Evaluate set union
 15 **for** $s = 1$ **to** k **do**
 16 $Itemp = \text{setdiff}(Itemp, I[s - 1, i])$ // Evaluate set difference
 17 set $I[k, i] = Itemp$
 18 **return** I

The accuracy of the mechanism \mathcal{M} at g_i is measured using the expected value of the distance $d(\mathcal{M}(g_i), f(g_i))$ as in Section 3.3, i.e.,

$$\mathbb{E}[d(\mathcal{M}(g_i), f(g_i))] = \sum_{r \in \mathcal{R}} p^{g_i}(r) |r - f(g_i)|, \quad (28)$$

where $f(g_i)$ denotes the number of triangles in the graph g_i , $p^{g_i}(r)$ denotes the probability of r in the mechanism \mathcal{M} when the input is g_i , and $\mathcal{R} = \{f(g_i) : g_i \in \mathcal{D}\}$. We compare the accuracy of our mechanisms to the Ladder mechanisms in [4, Algorithm 1]. For the fairness of comparison, we set the codomain of the counting function f in [4, Algorithm 1] be \mathcal{R} as above instead of \mathbb{Z} . Furthermore, we substitute 2ϵ for ϵ in [4, Algorithm 1] which ensures that the Ladder mechanism is 2ϵ -differentially private as ours. We evaluate the rate $Rate(g_i) = \frac{mean(g_i)}{mean'(g_i)}$, where $mean(g_i)$ denotes the expected value $\mathbb{E}[d(\mathcal{M}(g_i), f(g_i))]$ of our mechanism and $mean'(g_i)$ denotes the corresponding expected value at g_i of the Ladder mechanism.

The details of the experiments are as follows. We set \mathcal{G}_n be \mathcal{G}_7 , where $|\mathcal{D}| = 1044$ and $|\mathcal{R}| = 28$. The results are shown in Fig. 6 where the point i in the

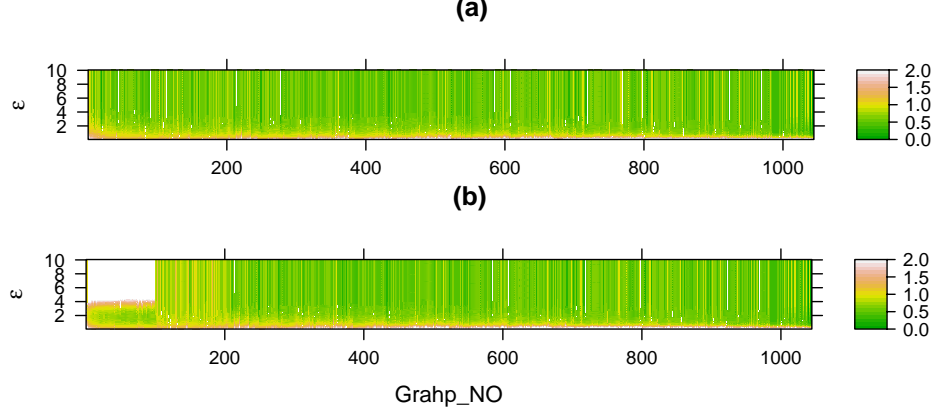


Fig. 6: The comparison of the mechanism in Theorem 3 and the Ladder mechanism of their mean values for \mathcal{G}_7

x -axis denotes the graph g_i , the point y in y -axis denotes the value ϵ and the value at the coordinate (i, y) is the value $Rate(g_i)$ when the input graph is g_i and $\epsilon = y$.

The upper figure in Fig. 6 shows the result when comparing the Ladder mechanism and the mechanism in Theorem 2. The below figure in Fig. 6 shows the result when comparing the Ladder mechanism and the mechanism in Theorem 3, where $\delta = 1$ for all the graphs in $\{g_i : i \in \{1, 2, \dots, 100\}\}$, and $\delta = 0$ for other graphs in \mathcal{D} . From Fig. 6 we can see the mechanism in Theorem 2 is better than the Ladder mechanism for most graphs when $\epsilon \geq 0.5$. However, the mechanism in Theorem 3 is worse than the Ladder mechanism for those graphs of $\delta = 1$ and for most ϵ . We reason that this is due to the (almost) monotonicity of the triangle counting function. This can be seen from the equation (25), where a_i^x 's raise more quickly than b_i^x 's when raising δ , in general.

6.2 Application in Linear Query

Linear query function (Definition 9) is a kind of well studied query functions in differential privacy [5, 12, 2, 3, 18]. Instead of treating batch linear queries, we treat a linear query.

We consider a special kind of the linear queries: the sum query. For the sum query, one dataset can be denoted as its histogram $x \in \mathbb{R}^N$, with x_i denoting the number of elements in x of type i [12, 17]. As discussed in Section 3.4 we use the neighboring set \mathcal{V}_f of f to denote the linear function f .

The details of the experiments are as follows. we consider four linear functions (over four different dataset universes) respectively. They are $\mathcal{V}_1 = [0, 1] \cup [1000, 1001]$, $\mathcal{V}_2 = [0, 100] \cup [1000, 1001]$, $\mathcal{V}_3 = [0, 500] \cup [1000, 1001]$, $\mathcal{V}_4 = [0, 1001]$, where $[a, b]$ denotes the corresponding interval in \mathbb{R} . Note that the first three sets are all concave set, which are different from the condition of the

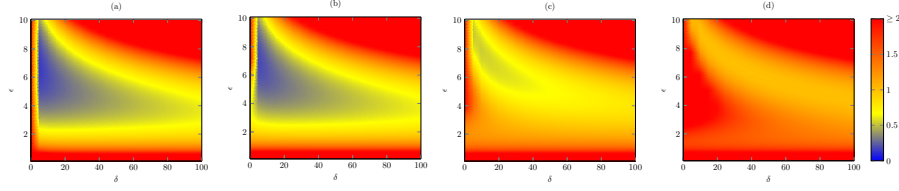


Fig. 7: The comparison of the mechanism in Theorem 3 and the Staircase mechanism of their mean values for queries $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4$ are shown in (a), (b), (c), (d), respectively

standard K -norm mechanism whose neighboring set K is convex. We implement the mechanisms in Theorem 2 and in Theorem 3 to these queries. The corresponding algorithms to evaluate the sequences $\{\mathcal{I}_i^x\}_i$ and $\{\mathcal{R}_i^x\}_i$ are similar with Algorithm 1 and are omitted. The main difference is that, since a linear query is symmetric for different datasets, each of the above two sequences is the same (when subtracting the corresponding $f(x)$) for different datasets. Therefore, we only need to evaluate the sequences for only one dataset. Note that the same value δ is assigned to different datasets due to the above symmetric property. Before giving the detailed experiments, we first present some theoretical results about linear queries.

Corollary 1. *The mechanisms in Theorem 2 and in Theorem 3 are ϵ -differentially private for the linear query.*

The above corollary is due to the symmetric property presented above which leads to $\alpha^x = \alpha^y$ (the proof of Theorem 2) for different datasets x, y .

We now discuss the convergence of the set sequences.

Definition 13 (The convergence of set sequence). *Let \mathcal{V}_f be a linear query function over \mathbb{R} . The corresponding set sequence $\{\mathcal{I}_i : i \in \bar{\mathbb{N}}\}$ is said to be convergent if there exist a_n and n such that $\mathcal{I}_n = a_n \pm [0, \Delta f]$ and $\mathcal{I}_{n+1} = a_n \pm [\Delta f, 2\Delta f]$, where Δf is the global sensitivity of \mathcal{V}_f .*

Proposition 3. *Assume $\mathcal{V}_f = [0, a] \cup [b, c]$ is a linear query function, where $0 < a < b < c$. Then the sequence $\{\mathcal{I}_i : i \in \bar{\mathbb{N}}\}$ of f is convergent when $i \geq \frac{\Delta f}{c-b}$.*

Proof. Note that the interval $[b, c]$ will generate the interval $[ib, ic]$ in \mathcal{I}_i^x . Setting $i \geq \frac{\Delta f}{c-b}$, we have $[(i-1)\Delta f, i\Delta f] \subseteq [ib, ic]$. This implies that $[i\Delta f, (i+1)\Delta f] \subseteq \mathcal{I}_{i+1}$. Then it is convergent.

The corresponding $\{\mathcal{R}_i : i \in \bar{\mathbb{N}}\}$ sequence has the similar convergence property with $\{\mathcal{I}_i : i \in \bar{\mathbb{N}}\}$ as in Proposition 3.

The accuracy of the mechanism \mathcal{M} is measured using the expected value of the distance $d(\mathcal{M}(x), f(x))$ as in Section 3.3, i.e.,

$$\mathbb{E}[d(\mathcal{M}(x), f(x))] = \int_{r \in \mathbb{R}} p(r) |r - f(x)| dr, \quad (29)$$

where $f(x) = 0$ and $p(r)$ denotes the probability of r in the mechanism \mathcal{M} when the input is x . We compare the accuracy of our mechanisms to the Staircase mechanism in [2, Algorithm 1]. We evaluate the rate $Rate = \frac{mean}{mean'}$, where $mean$ denotes the expected value $\mathbb{E}[d(\mathcal{M}(x), 0)]$ of the mechanism in Theorem 3 and $mean'$ denotes the corresponding expected value of the Staircase mechanism. The results are shown in Fig. 7 where the x -axis denotes the values of δ , the y -axis denotes the value of ϵ and the value at the coordinate (δ, ϵ) is the value $Rate = \frac{mean}{mean'}$ for the definite δ and ϵ . In Fig. 7, the results of the four queries $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4$ are shown in (a), (b), (c), (d), respectively.

We now analyze the results in Fig. 7. The four queries $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{V}_4$ have the same global (and local) sensitivity $\Delta f = 1001$. However, the volumes of their neighboring set (Section 3.4) are different. Explicitly, $\mu(\mathcal{V}_1) = 2, \mu(\mathcal{V}_2) = 101, \mu(\mathcal{V}_3) = 501, \mu(\mathcal{V}_4) = 1001$. The Fig. 7 shows some interesting phenomenon: The more larger of the value $\Delta f / \text{Vol}(\mathcal{V}_f)$, the more better of our mechanisms compared to the Staircase mechanism when $\epsilon \geq 3$ and $5 \leq \delta \leq 50$. Since, when $\delta = 0$, the mechanisms in Theorem 3 and in Theorem 2 are the same, where the K -norm mechanism is as a special case of the one in Theorem 2 as discussed in Section 4.1, we conclude that the K -norm mechanism (discrete version) is worse than the Staircase mechanism for the four queries as shown in Fig. 7. The K -norm mechanism works not well for the four linear functions, which is reasoned to be the non-monotonicity of the queries (at least for the first three queries) and which will generate concave sets K .

Another interesting phenomenon is that the density functions of the first three functions in Fig. 7 are not monotonic. This is counterintuitive since it is custom to assign less probability values to the points far away from $f(x)$, such as the Laplace density function or the Gaussian density function. Figure 8 shows the density functions of the corresponding mechanisms.

7 Related Work

As discussed in Section 1, the sensitivity-based methods are extensively used to construct differentially private mechanisms. These methods include the global sensitivity-based method [19,20], the smooth sensitivity-based method [7], the lo-

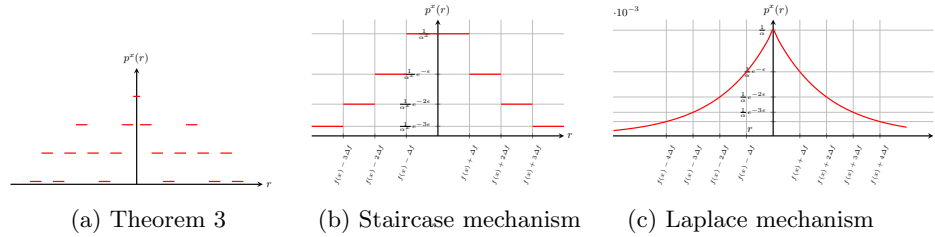


Fig. 8: The density functions of the mechanism in Theorem 3, the Staircase mechanism and the Laplace mechanism

cal sensitivity-based method [4] among others variants or approximations [8,9,10]. In section 4.1, we have shown that the sensitivity-based methods are just those using the metric d to construct mechanisms. Our mechanisms presented in Section 4 not just use the metric d , but also use the metric \bar{d} to construct mechanisms. Moreover, by using both d and \bar{d} , we have constructed six kinds of mechanisms and as the sensitivity-based methods a special case of these mechanisms. Furthermore, as shown in Section 4.1, the K -norm mechanism in [5,6] can also be considered as a special case of our mechanisms. The mechanisms in Theorem 2 and Theorem 3 for linear queries first appear in our paper [46]. However, this paper mainly focuses on proving the optimality of these mechanisms and does not realize the relation to the two metrics \bar{d} and d . To the best of our knowledge, this paper is the first to clearly point out that the sensitivity-based method mainly uses the metric d to construct mechanisms and is the first to realize that one can construct mechanisms by using the metric \bar{d} or by using both \bar{d} and d .

Traditionally, the Exponential mechanism [20] is seen as a universal mechanism for all query functions. However, the score function used in the Exponential mechanism is not presented explicitly, which means that one needs to design score functions when using the Exponential mechanism. However, how to design a good score function is not known. Our mechanisms presented in Section 4 can be considered as some special cases of the Exponential mechanism. For example, the mechanism in Theorem 2 uses the score function $q^x(r) = -\bar{d}(x, f^{-1}(r))$, which denotes the minimal distance (about metric \bar{d}) of x and the elements within the set $f^{-1}(r)$. On the other hand, our mechanisms can be considered as how to design score functions in the Exponential mechanism. Moreover, they are universal for all data processing problems.

The composition privacy property in Proposition 1 is a powerful property to construct mechanisms for complicated applications [47,25,32], which simplifies mechanism design in a modular construction way. Our model is consistent with the composition privacy property by using the product metric space and the product probability space as shown in Section 3.1.

The batch linear queries problems [48,18,49,39] are studied extensively in differential privacy. These works focus on the dependence (such as linear dependence) among the batch linear queries. In order to improve the accuracy, they first compress the high-dependent linear queries to the low-dependent or independent linear queries of small numbers and then use a differentially private mechanism, such as the Laplace mechanism, on the later, or as a whole. Our work can be used to treat the later low-dependent or independent linear queries to improve the accuracy of the whole of the batch linear queries.

There are a large amount of works [50,51,52,53] which approximate the batch linear queries to a dataset x by the batch linear queries to another dataset y , a synthetic dataset. These works are based on the statistical property of both the query functions and the arguments, i.e., the datasets. Our work currently does not treat these approximation problems.

There are a lot of works, which focus on the impossibility results [54] or the low bounds results [5,12,55] of the linear queries or the low-sensitivity queries. In this paper, we mainly focus on how to design universal mechanisms but do not evaluate impossibility results or low bounds of noise complexity, which would be one future work.

We model the differential privacy problem as finding a randomized mapping between two metric spaces. There are somewhat similar treatments about these concepts in [21,22]. Our treatment is different from theirs in the following aspect. The papers [21,22] mainly focus on either generalizing the differential privacy to a broaden scope or unifying the current problems or methods in differential privacy, whereas ours mainly focuses on finding methods to construct mechanisms.

Metric embedding [45] seems to have deep connection to differential privacy. The paper [56] uses the Johnson-Lindenstrauss transform to construct differentially private mechanism for some applications, such as PCA, Minimum cut query. The paper [57] uses a variant of Bourgain’s theorem to solve k -means distances problem in differential privacy. Our work abstracts the differential privacy problem as finding a randomized mapping between two metric spaces which seems similar with the randomized metric embedding problem [45]. The detailed discussions about their differences are presented in Section 3.6.

8 Conclusion and Future Work

In this paper, we discuss how to design differentially private mechanisms. Designing a good differentially private mechanism for a data processing problem is difficult since it needs to do a complicated balance between privacy and utility. (We seldom consider the efficiency of algorithms in this paper.) Designing a good universal mechanism which can adapt to many data processing problems is more difficult since the differential privacy model is task-specific, i.e., each data processing problem will have its own different mechanism(s). The sensitivity-based method is the first solution to design universal mechanisms, which employs the (global, local etc.) sensitivity of a query function as parameter to adapt to different data processing problems. In this paper, we extend the idea of the sensitivity-based method and find another methods to design universal mechanisms, which can adapt to different data processing problems by tuning a lot of parameters as shown in Section 5. The main observation of our mechanisms is to realize that the sensitivity-based method just employs the metric about utility to design mechanism. The heart of our mechanisms is to use either the metric about privacy or the metric about utility, or both, to design mechanisms.

Currently, we can only reinterpret the global sensitivity mechanism, the local sensitivity mechanism, the K -norm mechanism, among others by using our model as shown in Section 4. We hope, in future, there will be other mechanisms to be reinterpreted by our model or whose techniques can be employed by our model, such as the mechanisms based on Lipschitz functions [11], the mechanisms about synthetic datasets [50,51,52,53] etc.

One shortcoming of our mechanisms is the lack of noise bounds. How to use the techniques in [5,12,55,6] to estimate the noise bounds of our mechanisms is one future work, which needs to estimate the measures or to analyze the variations of those set sequences, such as $\{\mathcal{I}_i^x\}_i$ and $\{\mathcal{R}_i^x\}_i$. This is challenging since the techniques in [5,12,6] mainly treat linear queries whose neighboring set \mathcal{V}_f is convex, but non-convex cases are more common, such as those in Section 6 and those of non-linear queries. Maybe, the convergence property in Definition 13 would be helpful.

References

1. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 265–284, 2006.
2. Quan Geng and Pramod Viswanath. The optimal noise-adding mechanism in differential privacy. *IEEE Trans. Information Theory*, 62(2):925–951, 2016.
3. Quan Geng and Pramod Viswanath. Optimal noise adding mechanisms for approximate differential privacy. *IEEE Trans. Information Theory*, 62(2):952–969, 2016.
4. Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 731–745, 2015.
5. Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 705–714, 2010.
6. Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1269–1284, 2012.
7. Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84, 2007.
8. Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *PVLDB*, 4(11):1146–1157, 2011.
9. Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Analyzing graphs with node differential privacy. In *TCC*, pages 457–476, 2013.
10. Shixi Chen and Shuigeng Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013*, pages 653–664, 2013.
11. Sofya Raskhodnikova and Adam D. Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 495–504, 2016.

12. Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 351–360, 2013.
13. Moritz Hardt and Aaron Roth. Beating randomized response on incoherent matrices. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1255–1268, 2012.
14. Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 11–20, 2014.
15. Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *PVLDB*, 5(11):1364–1375, 2012.
16. Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: private data release via bayesian networks. In *SIGMOD Conference*, pages 1423–1434, 2014.
17. Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
18. Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor, and Vibhor Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.
19. Cynthia Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006.
20. Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.
21. Naoise Holohan, Douglas J. Leith, and Oliver Mason. Differential privacy in metric spaces: Numerical, categorical and functional data under the one roof. *Inf. Sci.*, 305:256–268, 2015.
22. Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings*, pages 82–102, 2013.
23. Erwin Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons. Inc, 1978.
24. Rui Chen, Gergely Ács, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM Conference on Computer and Communications Security*, pages 638–649, 2012.
25. Rui Chen, Noman Mohammed, Benjamin C. M. Fung, Bipin C. Desai, and Li Xiong. Publishing set-valued data via differential privacy. *PVLDB*, 4(11):1087–1098, 2011.
26. Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, pages 193–204, 2011.
27. Michael Hay, Chao Li, Gerome Miklau, and David D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM 2009, The Ninth IEEE International Conference on Data Mining, Miami, Florida, USA, 6-9 December 2009*, pages 169–178, 2009.
28. Cynthia Dwork, Moni Naor, Omer Reingold, and Guy N. Rothblum. Pure differential privacy for rectangle queries via private partitions. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application*

- of *Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 735–751, 2015.
29. Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 371–380, 2009.
 30. Cynthia Dwork, Weijie Su, and Li Zhang. Private false discovery rate control. *CoRR*, abs/1511.03803, 2015.
 31. Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 627–636, 2009.
 32. Noman Mohammed, Rui Chen, Benjamin C. M. Fung, and Philip S. Yu. Differentially private data release for data mining. In *KDD*, pages 493–501, 2011.
 33. Kamalika Chaudhuri, Daniel J. Hsu, and Shuang Song. The large margin mechanism for differentially private maximization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1287–1295, 2014.
 34. Kamalika Chaudhuri, Anand D. Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *Journal of Machine Learning Research*, 14(1):2905–2943, 2013.
 35. Kamalika Chaudhuri and Staal A. Vinterbo. A stability-based validation procedure for differentially private machine learning. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2652–2660, 2013.
 36. Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.
 37. Michael Kapralov and Kunal Talwar. On differentially private low rank approximation. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1395–1414, 2013.
 38. Vishesh Karwa, Sofya Raskhodnikova, Adam D. Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *ACM Trans. Database Syst.*, 39(3):22:1–22:33, 2014.
 39. Ziteng Wang, Kai Fan, Jiaqi Zhang, and Liwei Wang. Efficient algorithm for privately releasing smooth queries. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 782–790, 2013.
 40. Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 289–296, 2008.
 41. Rob Hall, Alessandro Rinaldo, and Larry A. Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(1):703–727, 2013.
 42. Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *Journal of Privacy and Confidentiality*, 4(1), 2012.

43. Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016.
44. Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 635–658, 2016.
45. Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 10–33, 2001.
46. Genqiang Wu, Xianyao Xia, and Yeping He. A near optimal differentially private mechanism for linear queries. *Ruan Jian Xue Bao/Journal of Software*, 28(9), 2017. (in Chinese) <http://www.jos.org.cn/1000-9825/5184.htm>.
47. Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 155–170, 2016.
48. Chao Li and Gerome Miklau. Optimal error of query sets under the differentially-private matrix mechanism. In *Joint 2013 EDBT/ICDT Conferences, ICDT '13 Proceedings, Genoa, Italy, March 18-22, 2013*, pages 272–283, 2013.
49. Grigory Yaroslavl'tsev, Graham Cormode, Cecilia M. Procopiuc, and Divesh Srivastava. Accurate and efficient private release of datacubes and contingency tables. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*, pages 745–756, 2013.
50. Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 339–356, 2012.
51. Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 61–70, 2010.
52. Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 765–774, 2010.
53. Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 2348–2356, 2012.
54. Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 71–80, 2010.
55. Anindya De. Lower bounds in differential privacy. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 321–338, 2012.
56. Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 410–419, 2012.

57. Zhiyi Huang and Aaron Roth. Exploiting metric structure for efficient private query release. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 523–534, 2014.